

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 6

Общее задание

Составить программу, в которой будет реализована работа с таблицей по варианту.

Пример решения

Обращаю внимание, что предложенные варианты являются не законченными решениями, а направлением действий, в котором можно двигаться для решения той или иной задачи.

```
#include <cstdlib>
#include <iostream>
#include <locale>
#include <cmath>
#include <ctime>
#include <iomanip>
#include <conio.h>

// *
// * -----
// * |                Буддийские монастыри Японии периода                |
// * -----
// * | Название | Школа | Количество монахов | Площадь земли (га) |
// * -----
// * | Тодайдзи | Т | 220 | 368,8 |
// * -----
// * | Якусидзи | С | 50 | 54,7 |
// * -----
// * | Дайандзи | Д | 10 | 12,2 |
// * -----
// * | Примечание: Т - Тэндай; С - Сингон; Д - Дзедзицу |
// * -----
// *

using namespace std;

int Size = 3; //Глобальная переменная, доступна в любом месте программы,
размер массива

class monastyr //Класс монастырь
{
public: //public - те данные класса, которые могут быть получены из любого
места программы
void add(string Name1, char School1, int NumMonks1, float Area1) //Функция
для ввода значений в конкретный экземпляр класса
```

```

{
    Name = Name1;
    School = School1;
    NumMonks = NumMonks1;
    Area = Area1;
}

string Name; // Элементы класса (бумажки в папке)
char School;
int NumMonks;
float Area;
};

monastyr* initialize() //Создание динамического массива с начальными
данными. Возвращает указатель на созданный массив
{
    monastyr* mon = new monastyr[Size]; //Объявляем массив из элементов
нашего класса

    mon[0].add("Тодайдзи", 'T', 220, 368.8); //Определяем каждый из элементов
массива.
    mon[1].add("Якусидзи", 'K', 50, 54.7);
    mon[2].add("Дайандзи", 'D', 10, 12.2);

    return mon; //Возвращаем указатель на массив
}

void editRow(monastyr* mon) //Изменение строки нашей таблицы
{
    int editIndex = 0; //Определяем переменную индекса для изменения
monastyr monToInsert; //Определяем объект монастыря, которым будем
заменять

    cout << "Введите номер строки, которую хотите изменить ";
    cin >> editIndex; //Получаем номер строки для изменения
editIndex--; //Учитываем, что номер строки на 1 больше индекса строки (
отсчет начинается с 0, а не с 1 )

    cout << "Введите данные о монастыре:" << endl;

    cout << "Название монастыря: ";
    cin >> monToInsert.Name; //Пользователь вводит Название монастыря и
прочие данные о нем

    cout << "Монастырь принадлежит к школе: ";

```

```

cin >> monToInsert.School;

cout << "Количество монахов: ";
cin >> monToInsert.NumMonks;

cout << "Площадь монастыря: ";
cin >> monToInsert.Area;

mon[editIndex] = monToInsert; //Заменяем объект по данному индексу на
НОВЫЙ
}

monastyr* insertRow(monastyr* mon) //Ввод новой строки в конец массива
{
    monastyr monToInsert; //Определяем объект монастыря, который будем
добавлять

    cout << "Введите данные о монастыре:" << endl;

    cout << "Название монастыря: ";
    cin >> monToInsert.Name; //Пользователь вводит Название монастыря и
прочие данные о нем

    cout << "Монастырь принадлежит к школе: ";
    cin >> monToInsert.School;

    cout << "Количество монахов: ";
    cin >> monToInsert.NumMonks;

    cout << "Площадь монастыря: ";
    cin >> monToInsert.Area;

    monastyr* NewArray = new monastyr[Size + 1]; //Определяем массив,
который на 1 объект больше, чем текущий, чтобы вставить новый объект в
конец

    for (int i = 0; i < Size; i++) //Пробегаем по всем значениям старого
массива и присваиваем новому массиву эти значения
    {
        NewArray[i] = mon[i];
    }

    NewArray[Size] = monToInsert; //Добавляем новый объект в конец
НОВОГО массива

```

```
delete[] mon; //Удаляем старый массив
```

```
Size++; //Поскольку массив стал больше, увеличиваем на 1 глобальную переменную с размером массива
```

```
return NewArray; //Возвращаем ссылку на новый массив. Функцию обязательно нужно использовать так: a = insertRow(a). Если этого не сделать, в a останется ссылка на старый массив, который мы благополучно удалили несколькими строчками выше  
}
```

```
void printRow(monastyr mon) //Напечатать одну строчку таблицы  
{  
    cout << " | " << mon.Name << setw(15 - mon.Name.length()) //Создаем красивый вывод для таблицы. От названия храма мы должны отступить одно и то же расстояние в каждой строчке.  
        << " | " << mon.School //Если имя содержит один символ, мы отступим 14 символов, если два - 13 и т.д.  
        << " | " << setw(19) << mon.NumMonks  
        << " | " << setw(19) << mon.Area  
        << " | " << endl;  
    cout << " -----" << endl;  
}
```

```
void printHeader() //Вывести заголовок  
{  
    cout << " -----" << endl;  
    cout << " |          Буддийские монастыри Японии периода          |" << endl;  
    cout << " -----" << endl;  
    cout << " | Название | Школа | Количество монахов | Площадь земли (га) |" << endl;  
    cout << " -----" << endl;  
}
```

```
void printTable(monastyr* mon) //Вывести таблицу  
{  
    system("CLS"); //Очистка терминала  
    printHeader(); //Печатаем заголовок  
  
    for (int i = 0; i < Size; i++) //Выводим все строчки  
    {
```

```

        printRow(mon[i]);
    }
}

monastyr* deleteRow(monastyr* mon) //Удаление строки
{
    int DeleteIndex = 0;
    cout << "Введите номер строки, которую хотите удалить ";
    cin >> DeleteIndex;
    DeleteIndex--; //Преобразуем номер строки в индекс

    monastyr* NewArray = new monastyr[Size - 1]; //Создаем новый массив,
на один объект меньше предыдущего

    for (int i = 0; i < Size; i++) //Пробегаем по всем объектам старого массива
    {
        if (i < DeleteIndex) //Если мы еще не дошли до удаляемого объекта,
записываем старые объекты в новый массив
        {
            NewArray[i] = mon[i];
        }
        else if (i > DeleteIndex) //Если мы перешли удаляемый объект,
продолжаем записывать, но изменяем индекс для записи
        {
            NewArray[i - 1] = mon[i];
        }
    }

    delete[] mon; //Удаляем старый массив

    Size--; //Мы уменьшаем размер массива, потому что он уменьшился

    return NewArray; //Возвращаем ссылку на новый массив, по аналогии с
добавлением строк
}

int main(int argc, const char* argv[])
{

    setlocale(LC_ALL, "Russian");

    monastyr* mon = initialize(); //Функция инициализации массива
начальными значениями

    printTable(mon); //Вывод таблицы на экран. Перед этим он очищается

```

mon = insertRow(mon); //Ввод новой строки, необходимо присваивать mon эту функцию, потому что в результате создается новый массив с новой ссылкой на него

```
printTable(mon);
```

```
editRow(mon); //Редактировать строку
```

```
printTable(mon);
```

mon = deleteRow(mon); //Удалить строку, то же, что и с добавлением строк

```
printTable(mon);
```

```
return 0;
```

```
}
```